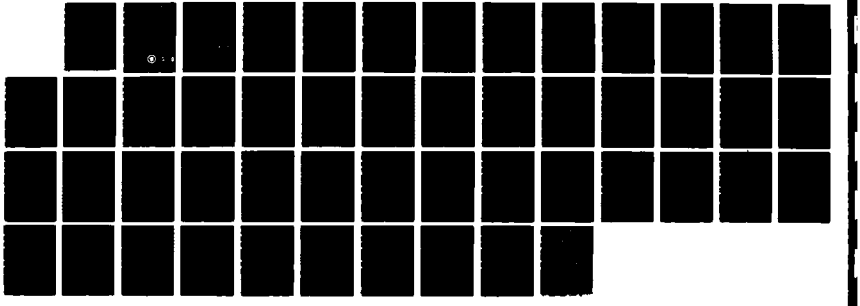
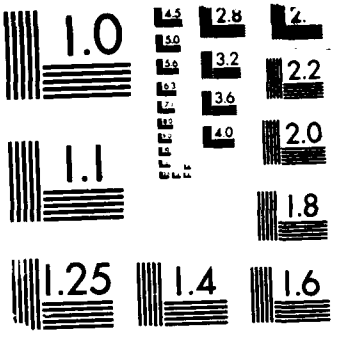
AD-A191 823 WFEMAS: A NONLINEAR ADAPTIVE FINITE ELEMENT SOLVER PART 1/1
1(U) PITTSBURGH UNIV PA INST FOR COMPUTATIONAL
MATHEMATICS AND APPLICATIONS C K MESZTENYI ET AL
UNCLASSIFIED DEC 87 ICMA-87-113 N00014-80-C-0455 F/G 12/1 NL





INSTITUTE FOR COMPUTATIONAL MATHEMATICS AND APPLICATIONS

ICMA-87-113 December 1987

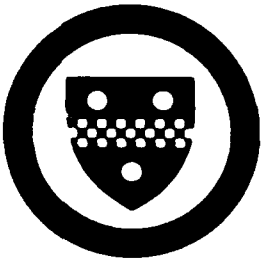
NFEARS
A Nonlinear Adaptive Finite Element Solver [1]

by

Charles K. Mesztenyi[2] and Werner C. Rheinboldt [3]

Department of Mathematics and Statistics

University of Pittsburgh



(4)

ICMA-87-113 December 1987

NFEARS
A Nonlinear Adaptive Finite Element Solver [1]

by

Charles K. Mesztenyi[2] and Werner C. Rheinboldt [3]

[1] This work was in part supported by the Office of Naval Research under contracts N-00014-80-C-9455 and N-00014-85-K-0169 and by the National Science Foundation under grant DCR-8309926. Acknowledgement is also made for the partial support of the Computer Science Center of the University of Maryland.

[2] Computer Science Center, University of Maryland, College Park, MD 20742

[3] Department of Mathematics and Statistics, University of Pittsburgh, Pittsburgh, PA 15260

Contents

Preface

The two parts of this report present an introduction to the design and usage of the NFEARS system developed jointly at the Universities of Maryland and Pittsburgh. The acronym stands for "Nonlinear Finite Element Adaptive Research Solver" and refers to an experimental software system for the solution of a class of nonlinear, stationary boundary value problems in two space dimensions involving two real parameters. The overall concept of the system is similar to that of the earlier system FEARS (see, e.g., [16],[17],[26])[1] for linear elliptic problems and retains many of the features of that system.

The eight sections of Part I. describe the mathematical background of the processes incorporated into NFEARS, and the six sections of Part II. serve as User's Manual for the program.

As the earlier program, NFEARS was developed strictly as a research tool and is expected to be subject to continual modifications.

[1] All references are collected in Section I.9 of Part I.

Introduction

As noted in the Preface, NFEARS is an experimental software system for the solution of a class of nonlinear, stationary boundary value problems in two space dimensions involving several parameters. The specific form of the problem class that can be handled by NFEARS will be given in the next Section. Generically, these are equations of the form

$$F(z,\lambda) = 0$$

involving a particular nonlinear mapping $F: X \rightarrow Y$. Here Y is a suitable function space and X is the product $X = Z \times \Lambda$ of another function space Z (the state space) and a finite-dimensional parameter space Λ. The basic formulation of the NFEARS problems incoporates a seven-dimensional parameter space, but for the calculation only a two-dimensional subspace Λ is used. Parameterized nonlinear equations arise in numerous areas. The applications we had especially in mind derive from nonlinear structural mechanics, but the system,of course, is more generally applicable.

In general, the set of all solutions (z,λ) of (1) forms a differentiable manifold M in the space X with a dimension equal to that of the parameter space Λ (see e.g. [19]). In most applications interest centers not so much on computing a few solutions (z,λ) for specific values of the parameters, but rather on analysing the form and special features of the manifold. For example, in the mentioned structural problems we may wish to determine those solutions where the stability behavior changes.

For the computation, a finite element approximation is applied to the basic equations. Since the parameter-vector does not need to be discretized, the resulting equations then have the generic form

$$F_h(z_h,\lambda) = 0$$

where now F_h maps a space $X_h = Z_h \oplus \Lambda$ into a space Y_h where now both Z_h and Y_h are finite dimensional. Since the parameter space is unchanged, we may expect that the solutions (z_h,λ) of the discretized equation form a differentiable

manifold M_h in X_h of the same dimension as the original manifold M in X. Moreover, we may imbed X_h into X which enables us to define the discretization error as a suitably specified distance between M and M_h (see [12],[13]). This allows, In turn, for the design of a posteriori estimates of this discretization error (see e.g. [6],[8],[18],[20]). As in the linear case (see e.g. [2],[7],[8],[10]) such a posteriori estimates are not only important in providing some assessment of the reliability of the computed results, but also in controlling an adaptive mesh-refinement procedure with the aim of obtaining a solution with appropriately bounded error behavior using minimal cost (see [1], [3]).

The basic procedures for the computational analysis of the solution manifold M_h are the continuation methods, or incremental methods, as they are also called in the engineering literature. When the dimension of the parameter space, and hence of M_h, exceeds one, these methods require an a priori restriction to some path on M_h and then produce a sequence of points along that path. In structural problems the parameters often define the load configuration and the path is defined by fixing the load points and load directions and leaving only a load intensity as scalar parameter.

Obviously, it is, in general, not easy to develop a good picture of a multi-dimensional manifold solely from information along such paths. This led recently to the development of methods for the computation of simplicial approximations of open subsets of M_h (see [21],[22]).

The goal of NFEARS is to provide a tool for the above listed tasks. In fact our generic discussion so far identifies all the principal capabilities of the system which can be summarized as follows:

(i) Construct a finite element discretization of the given equations using biquadratic elements on a hierarchical class of meshes defined recursively by repeated refinement or de-refinement.

(ii) With a given solution x_{h0} as starting point and a path through x_{h0} specified by a selected combination of the parameters, apply a continuation process to compute a sequence of points approximating that path. The process follows the

design of the PITCON system (see [23],[24]). It allows also for the computation of specific target points and limit points on the path.

(iii) At selected computed points along the continuation path, compute a posteriori estimators for the discretization error.

(iv) On the basis of a "density" and "intensity" concept for finite element meshes, use the error estimators to control the modification of the current mesh.

(v) At any of the computed points, apply an algorithm for the computation of a simplicial approximation of an open region of the manifold M_h surrounding the particular point.

I.1. Problem Class

The nonlinear problems underlying NFEARS are formulated in a weak form requiring the determination of a function $U \in X$ such that

$$B(U,V) = L(V) , \quad \forall \ V \in Y,$$

where X,Y are appropriate function spaces which shall not be further identified. Here B is a bivariate form

$$B(U,V) = \int_{\Omega} \frac{\partial \Phi}{\partial U}(\sigma_1,x,y,U,U_x,U_y)\ V\ dxdy +$$

$$\int_{\Omega} \frac{\partial \Phi}{\partial U_x}(\sigma_1,x,y,U,U_x,U_y)V_x dxdy + \int_{\Omega} \frac{\partial \Phi}{\partial U_y}(\sigma_1,x,y,U,U_x,U_y)V_y dxdy$$

and L a linear functional

$$L(V) = \int_{\Omega} G_2(\sigma_2, x, y)\ V\ dxdy + \int_{\Gamma} G_1(\sigma_3, x, y)\ V\ d\gamma$$

involving the given functions

$$(\sigma_1,x,y,U^0,U^1,U^2) \rightarrow \Phi(\sigma_1,x,y,U^0,U^1,U^2) \ , \ \forall\, \sigma_1 \in R^2, x,y, U^0,U^1,U^2 \in R^1$$

$$(\sigma_2,x,y) \rightarrow G_2(\sigma_2,x,y), \qquad \forall\, \sigma_2 \in R^2, \ x,y \in R^1$$

$$(\sigma_3,x,y) \rightarrow G_1(\sigma_3,x,y), \quad \forall\, \sigma_3 \in R^2 \ , \ x,y \in R^1$$

In this formulation we use the notation:

- U the trial function in X,
- V the test function in Y,
- Ω a given domain in R^2 which will be described in more detail in Section I.2,
- Γ a given subset of $\partial\Omega$ which will also be discussed in Section I.2

x,y the global coordinates in R^2

γ the arclength of Γ,

σ_i i=1,2,3, three two-dimensional parameter vectors which will be discussed below.

The trial and test spaces X and Y are assumed to allow us to handle prescribed Dirichlet boundary conditions of the form

$$U = \begin{cases} b(x,y), & \text{on } \Theta_1 \\ \sigma_4\, b(x,y), & \text{on } \Theta_2 \end{cases} \qquad \Theta_1 \cup \Theta_2 = \Theta$$

where Θ is a given one-dimensional subset of Ω, σ_4 a scalar parameter, and b a function which is locally, piecewise quadratic (see Section I.3).

All given functions are supposed to be sufficiently smooth on their respective domains. For the computation it is required that subroutines are available for the computation of the following functions

$$\frac{\partial \Phi}{\partial U^k}\ ,\ \frac{\partial^2 \Phi}{\partial \sigma_1 \partial U^k}\ ,\ \frac{\partial^2 \Phi}{\partial U^k\, \partial U^j}\ , \qquad k,j = 0,\ 1,\ 2$$

$$G_2\ ,\ \frac{\partial G_2}{\partial \sigma_2}\ ,\ G_1\ ,\ \frac{\partial G_1}{\partial \sigma_3}$$

and it is assumed that appropriate values are provided which define the boundary function b(x,y).

Altogether the problem involves a seven-dimensional space of parameters $(\sigma_1,\sigma_2,\sigma_3,\sigma_4) \in R^2 \times R^2 \times R^2 \times R^1$. Thus in general, the set of all solutions U is expected to be a seven dimensional differentiable manifold in the space X (see e,g, [19]). In applications, interest rarely centers on the determination of a few specific solutions of this problem for fixed parameter-values, but rather on an assessment of the behavior of these solutions when the parameters vary; that is, on an analysis of the form and features of the solution manifold. NFEARS is intended to provide a tool for such an analysis.

Evidently, serious data handling problems are likely to arise in an analysis of a seven dimensional manifold, even though, in principle, the computational methods, used here for approximating segments of such manifolds, remain applicable. Accordingly, for the computation, NFEARS is restricted to two-dimensional sub-manifolds. These sub-manifolds are defined by linear combinations of the three parameter vectors σ_i ,i=1,2,3, and of the scalar parameter σ_4 in dependence of two effective parameters λ_1 and λ_2. More specifically, the following linear relations between the σ_i and λ_j are assumed to be given:

$$\sigma_i = \begin{bmatrix} \sigma_i^1 \\ \sigma_i^2 \end{bmatrix} = \begin{bmatrix} \alpha_i^1 \\ \alpha_i^2 \end{bmatrix} + \begin{bmatrix} \beta_i^1 & 0 \\ 0 & \beta_i^2 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}, \quad i=1,2,3$$

$$\sigma_4 = \alpha_4 + \beta_4 \lambda_1$$

Accordingly, we are now interested in the two dimensional manifold of solutions $(U,\lambda_1,\lambda_2) \in X \times R^2$ of our problem, and NFEARS computes finite element approximations of points along paths as well of nodes of triangulations of open subsets of this manifold.

I.2. Domains and Mappings

The domain W of the problem is assumed to be sub-divided by the user into a collection of sub-domains. The interior of these sub-domains are called 2-D domains and will be denoted here by

$$\Omega_k^2 , \quad k = 1,...,N_2$$

The closure of each 2-D domain is a generalized quadrilateral with four corner points and four sides each of which can be either a straight line or a circle. The (relatively) open line segments forming the sides are called 1-D domains (or lines, for short) and are denoted by

$$\Omega_k^1 , \quad k = 1,...,N_1$$

Finally, we call the corner points 1-D domains (or points for short) and denote them by

$$\Omega_k^0 , \quad k = 1,..,N_0$$

Thus altogether the original domain is decomposed as follows

$$\Omega = (\bigcup_{k=1}^{N_2} \Omega_k^2) \cup (\bigcup_{k=1}^{N_1} \Omega_k^1) \cup (\bigcup_{k=1}^{N_0} \Omega_k^0)$$

The closure of each 2-D domain is the homeomorphic image of the unit square $S = \{(\xi,\eta); 0 \le \xi,\eta \le 1\}$ in the plane where all computations are performed. The specific form of the mapping is given below. Clearly, these transformations onto S require that degeneracies in the definition of any 2-D domain Ω_k^2 must be avoided. In particular, the angles formed at the corners of Ω_k^2 should not be too close to 0 or π, no overlapping sides should occur, and the overall shape should not be approximately triangular. Figure I.2.1 shows some illegal and legal configurations.

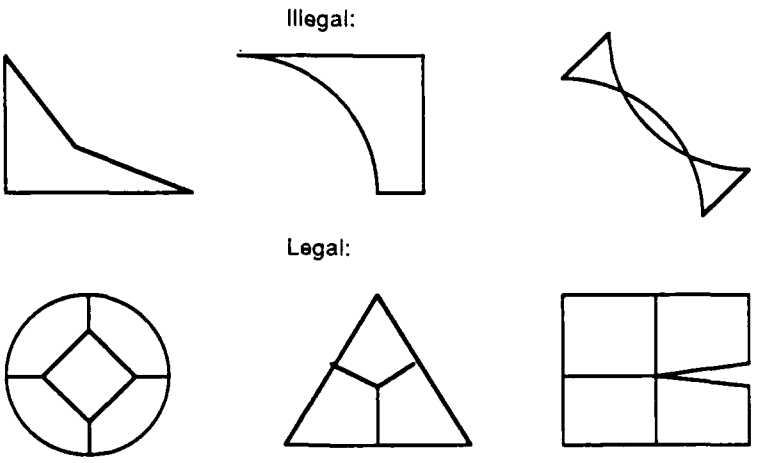


Figure I.2.1

Cracks may be introduced by using two 1-D domains between two 2-D domains. If the crack originates at the external boundary of Ω, then two 0-D domains with the same coordinates should be used :

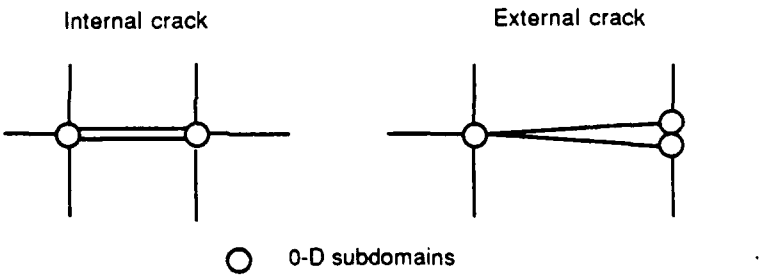


Figure I.2.2

As noted before , the allowable 1-D domains are either straight line segments or circular arc segments with a given radius. More, specifically, any such domain Ω_k^1 is defined as a directed arc from a starting point with coordinates (x_1,y_1) to an endpoint with coordinates (x_2,y_2). Of course, both of these points are 0-D domains. The order of these two bounding 0-D domains defines a direction for Ω_k^1 which is used to set the direction of the unit tangent vector t at the points of the arc. The corresponding unit normal vectors n are obtained by rotating t counter- clockwise through an angle $\pi/2$ (see Figure I.2.3).

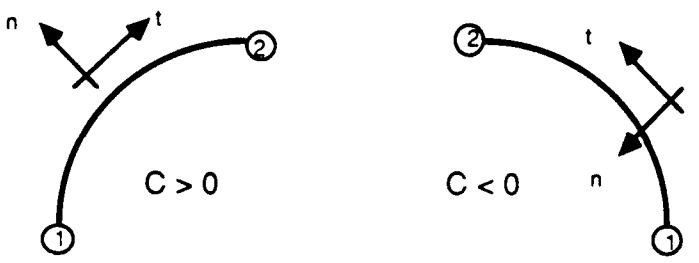


Figure I.2.3

The particular form of Ω_k^1 is specified by its signed curvature C_k^1. More specifically, if $C_k^1 = 0$, then Ω_k^1 is a straight line segment, else it is a circular arc with radius $1/|C_k^1|$. In the latter case, if $C_k^1>0$ or $C_k^1<0$, then the center of the circle is to the right or the left of the arc, respectively, when looking in its direction (see Figure I.2.3).

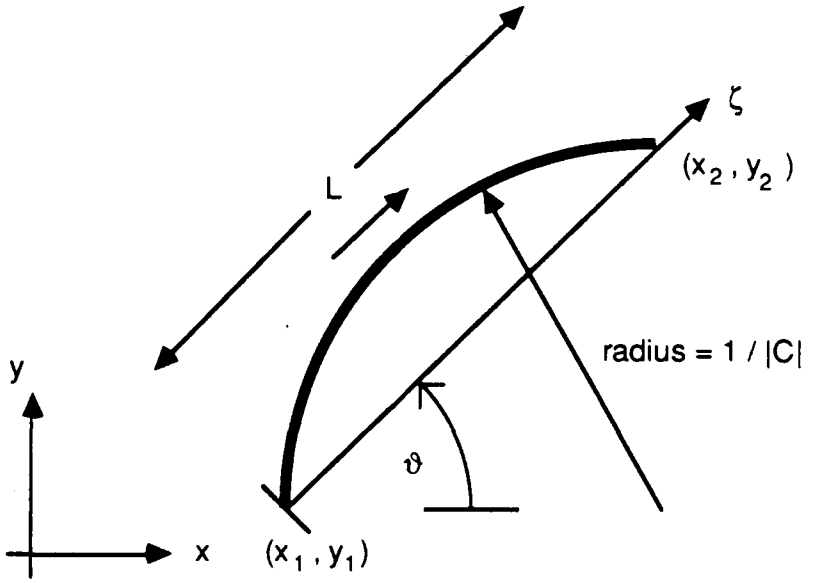


Figure I.2.4

Hence the tangent and normal vector at any point of the arc are given by

$$t = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} c_\zeta \cos\vartheta + s_\zeta \sin\vartheta \\ c_\zeta \sin\vartheta - s_\zeta \cos\vartheta \end{bmatrix}, \qquad n = \begin{bmatrix} -t_2 \\ t_1 \end{bmatrix}$$

where

$$\cos\vartheta = \frac{x_2 - x_1}{L}, \quad \sin\vartheta = \frac{y_2 - y_1}{L}$$

$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \qquad \rho = \frac{L}{2}C$$

$$s_\zeta = \rho(2\zeta - 1), \quad c_\zeta = \sqrt{1 - \rho^2(2\zeta - 1)^2}$$

Accordingly, the global coordinates (x,y) and the local coordinates (ζ,η) are related as follows

$$\begin{bmatrix} x(\zeta) \\ y(\zeta) \end{bmatrix} = \frac{1}{2}\begin{bmatrix} x_1 + x_2 \\ y_1 + y_2 \end{bmatrix} + \frac{L}{2}\begin{bmatrix} \cos\vartheta & -\sin\vartheta \\ \sin\vartheta & \cos\vartheta \end{bmatrix}\begin{bmatrix} 2\zeta - 1 \\ \rho\dfrac{1-(2\zeta-1)^2}{\sqrt{1-\rho^2} + \sqrt{1-\rho^2(2\zeta-1)^2}} \end{bmatrix}$$

Line integration of a function f along the 1-D domain then becomes

$$\int_{\Omega_\kappa^1} f\,d\gamma = L\int_0^1 \frac{f}{c_\zeta}\,d\zeta$$

Transformation of a 2-D domain is defined by a Coon's patch; that is, by blending the transformation functions (x(k),y(k)) , k=1,2,3,4 of the four bounding 1-D domains. As Figure I.2.5 shows, the 0-D and 1-D domains are indexed in a counter clockwise direction.

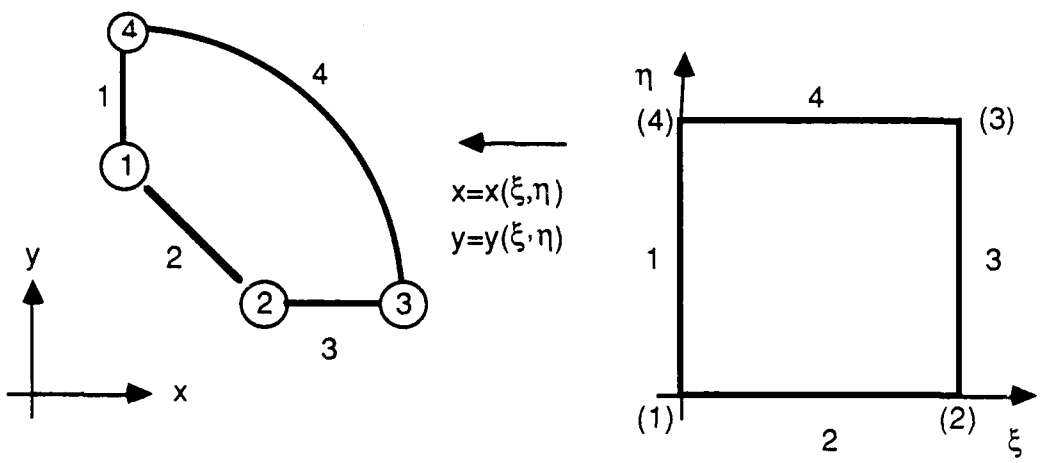


Figure I.2.5

The transformation functions can be expressed in the following form:

$$x(\xi,\eta) = \begin{bmatrix} 1-\xi & 1 & \xi \end{bmatrix} \begin{bmatrix} -x_1^0 & x^{(1)}(\eta_1) & -x_4^0 \\ x^{(2)}(\xi_2) & 0 & x^{(4)}(\xi_4) \\ -x_2^0 & x^{(3)}(\eta_3) & -x_3^0 \end{bmatrix} \begin{bmatrix} 1-\eta \\ 1 \\ \eta \end{bmatrix}$$

$$y(\xi,\eta) = \begin{bmatrix} 1-\xi & 1 & \xi \end{bmatrix} \begin{bmatrix} -y_1^0 & y^{(1)}(\eta_1) & -y_4^0 \\ y^{(2)}(\xi_2) & 0 & y^{(4)}(\xi_4) \\ -y_2^0 & y^{(3)}(\eta_3) & -y_3^0 \end{bmatrix} \begin{bmatrix} 1-\eta \\ 1 \\ \eta \end{bmatrix}$$

Here (x_i^o, y_i^o), i=1,2,3,4, are the coordinates of the four 0-D domains and $(x^{(k)}(\eta_k), y^{(k)}(\eta_k))$, k=1,3, $(x^{(k)}(\zeta_k), y^{(k)}(\zeta_k))$, k=2,4, the four 1-D transformation functions. Their arguments η_k, or ζ_k reflect the appropriate orientation of the corresponding 1-D domains; that is,

$$\eta_k = \iota_k (\eta_k + \frac{\iota_k - 1}{2}) \quad , \quad \zeta_k = \iota_k (\zeta_k + \frac{\iota_k - 1}{2})$$

where $\iota_k = +1$, if the orientation of the k-th 1-D domain corresponds to the orientation of the y-axis or x-axis, respectively, and $\iota_k = -1$ if it is the opposite. Since the determinant of the Jacobian of this transformation appears in each volume integral, it is important to express it in an efficient form for later evaluation. The partial derivatives can be written as

$$\frac{\partial}{\partial \xi} \begin{bmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{bmatrix} = \begin{bmatrix} r_2(\xi) \sin \vartheta_2 & x_3^*(\eta) - x_1^*(\eta) & r_4(\xi) \sin \vartheta_4 \\ r_2(\xi) \cos \vartheta_2 & y_3^*(\eta) - y_1^*(\eta) & r_4(\xi) \cos \vartheta_4 \end{bmatrix} \begin{bmatrix} 1 - \eta \\ 1 \\ \eta \end{bmatrix}$$

$$\frac{\partial}{\partial \eta} \begin{bmatrix} x(\xi,\eta) \\ y(\xi,\eta) \end{bmatrix} = \begin{bmatrix} r_1(\eta) \sin \vartheta_1 & x_4^*(\xi) - x_2^*(\xi) & r_3(\eta) \sin \vartheta_3 \\ r_1(\eta) \cos \vartheta_1 & y_4^*(\xi) - y_2^*(\xi) & r_3(\eta) \cos \vartheta_3 \end{bmatrix} \begin{bmatrix} 1 - \xi \\ 1 \\ \xi \end{bmatrix}$$

where

$$r_k(\zeta) = L_k \frac{\rho_k (2\zeta - 1)}{\sqrt{1 - \rho_k^2 (2\zeta - 1)^2}} \quad , \quad p_k(\zeta) = \rho_k \frac{1 - (2\zeta - 1)^2}{\sqrt{1 - \rho_k^2} + \sqrt{1 - \rho_k^2 (2\zeta - 1)^2}}$$

with

$$\begin{bmatrix} x_k^*(\zeta) \\ y_k^*(\zeta) \end{bmatrix} = \begin{bmatrix} x_k^m \\ y_k^m \end{bmatrix} + \frac{L_k}{2} \begin{bmatrix} \cos \vartheta_k & -\sin \vartheta_k \\ \sin \vartheta_k & \cos \vartheta_k \end{bmatrix} \begin{bmatrix} \iota_k (2\zeta - 1) \\ p_k(\zeta) \end{bmatrix}$$

$$\begin{bmatrix} x_k^m \\ y_k^m \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_k^0 + x_{k-1}^0 \\ y_k^0 + y_{k-1}^0 \end{bmatrix} \quad k = 2,3,4 \quad \text{and} \quad \begin{bmatrix} x_1^m \\ y_1^m \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_1^0 + x_4^0 \\ y_1^0 + y_4^0 \end{bmatrix}$$

I.3. Finite Element Approximation

The mesh on our domain Ω consists of curvilinear elements which are first defined on the basic square $S = \{ (\xi,\eta) , 0 \le \xi,\eta \le 1\}$ and then mapped onto the closures of the 2-D domains which make up Ω. A nine-node planar Lagrangian quadratic element is used as the master element of NFEARS.

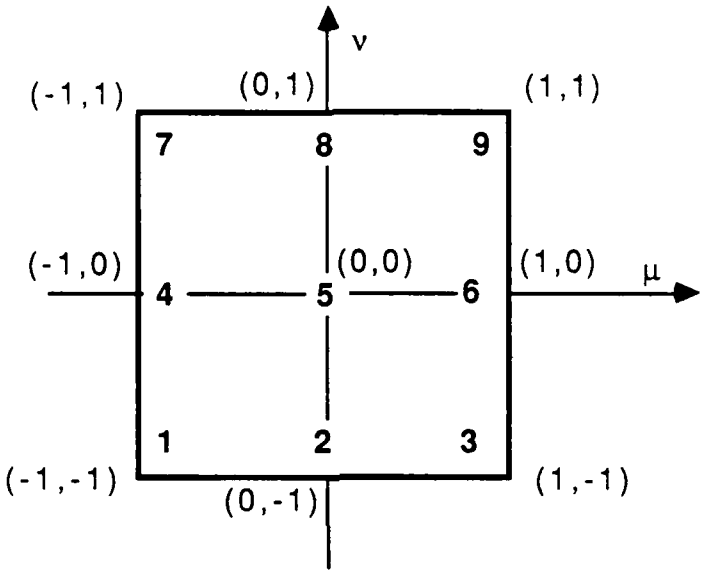


Figure I.3.1

The shape functions associated with its nine nodes (see Figure I.3.1) are as follows:

$$\varphi_1 = \frac{1}{4}\mu(\mu-1)\nu(\nu-1)\ ;\ \varphi_2 = \frac{1}{2}(1-\mu^2)\nu(\nu-1)\ ;\ \varphi_3 = \frac{1}{4}\mu(\mu+1)\nu(\nu-1)$$

$$\varphi_4 = \frac{1}{2}(1-\nu^2)\mu(\mu-1);\ \varphi_5 = (1-\mu^2)(1-\nu^2)\ ;\ \varphi_6 = \frac{1}{2}(1-\nu^2)\mu(\mu+1)$$

$$\varphi_7 = \frac{1}{4}\mu(\mu-1)\nu(\nu+1);\ \varphi_8 = \frac{1}{2}(1-\mu^2)\nu(\nu+1);\ \varphi_9 = \frac{1}{4}\mu(\mu+1)\nu(\nu+1)$$

As in [26] the admissible meshes on S are defined recursively by the two rules :

(a) The mesh consisting of the four quarter-squares

$$S_k = \{(\xi,\eta)\ ;\ i \le 2\xi \le i+1,\ j \le 2\eta \le j+1\ \},\quad k = 1+j+2i,\ i,j=0,1$$

of S is admissible. These four squares S_1, S_2, S_3, S_4 are called superelements.

(b) If Δ is an admissible mesh on S, then the mesh is admissible that is obtained from Δ by quartering any one square s of Δ into four congruent squares of half the side-length of s.

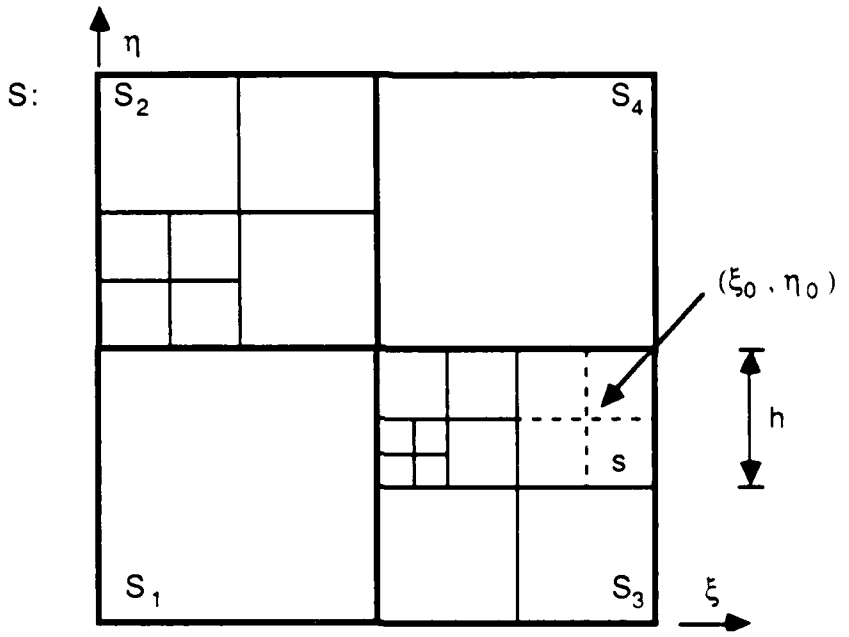


Figure I.3.2

Figure I.3.2 shows an example of an admissible mesh on S. Any element s of an admissible mesh Δ on S is uniquely characterized by its center point (ξ_0, η_0) and side-length h , and the transformation

$$\mu \Rightarrow \xi = \xi_0 + (0.5h)\,\mu\,, \qquad \nu \Rightarrow \eta = \eta_0 + (0.5h)\,\nu$$

maps the master element onto s. This specifies the shape functions on s. Instead of detailing them, we give here the more general formulas for quadratic interpolation on s. As above let

$$\mu = \frac{2}{h}(\xi - \xi_0) \quad , \quad \nu = \frac{2}{h}(\eta - \eta_0)$$

and suppose that u_i, i=1,2,...,9 are given scalar values at the nine nodes of s indexed in the same order as in Figure I.3.1. Then the interpolated value $u(\xi,\eta)$ at the point $(\xi,\eta) \in s$ is given by

$$u(\xi,\eta) = \left[\mu(\mu-1),\ 1,\ \mu(\mu+1)\right] \begin{bmatrix} u_1-u_2-u_4+u_5 & 2(u_4-u_5) & u_7-u_8-u_4+u_5 \\ 2(u_2-u_5) & 4u_5 & 2(u_8-u_5) \\ u_3-u_2-u_6+u_5 & 2(u_6-u_5) & u_9-u_8-u_6+u_5 \end{bmatrix} \begin{bmatrix} \nu(\nu-1) \\ 1 \\ \nu(\nu+1) \end{bmatrix}$$

It will be useful to record also the corresponding quadratic interpolation which is induced on any line segment $[\zeta_0-h/2\ ,\ \zeta_0+h/2]$. If u_i, i=1,2,3 are the values at the three nodes (indexed in order of increasing ζ-values), then the value $u(\zeta)$ of the one-dimensional quadratic interpolation is specified by

$$u(\zeta) = \left[\mu(\mu-1),\ 1,\ \mu(\mu+1)\right] \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad \mu = \frac{2}{h}(\zeta - \zeta_0)$$

with $\zeta = \xi$ or η.

Once the shape functions are defined on a given admissible mesh on S, we have obtained, at the same time, the corresponding space of conforming finite element functions on that mesh. Hence, by applying the transformations from S onto each one of the closed 2-D domains Ω_k^2, we construct a curvilinear mesh on the full domain Ω. This is schematically indicated in Figure I..3 for the basic mesh of four superelements.

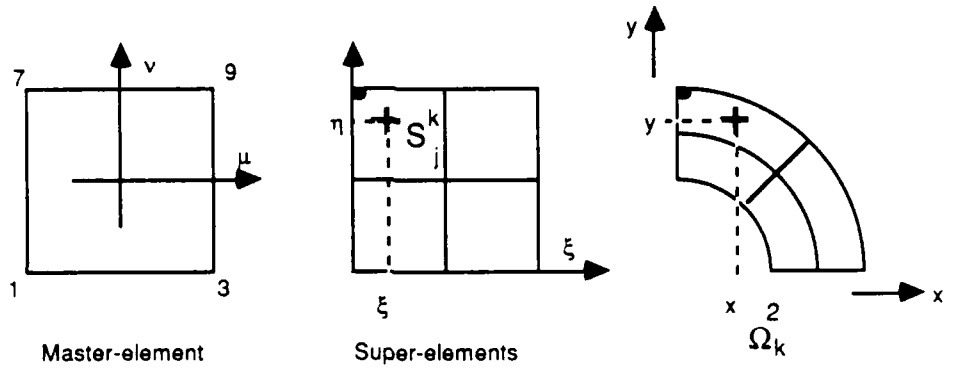


Figure I.3.3

Let $v_i = v_i(x,y)$ denote the global shape function belonging to the node i of the resulting mesh on Ω. Then the finite element space, to be considered here, consists of all the functions of the form

$$u(x,y) = \sum_{i=1}^{N} u_i v_i(x,y) \ , \quad x,y \in \Omega$$

where the index N denotes the total number of "free" nodes; that is, of the nodes not contained in the 0-D and 1-D domains carrying Dirichlet boundary condition.

Dirichlet conditions can be prescribed on a set Θ consisting of 0-D and 1-D domains, and they may either depend linearly on σ_4 or be independent of that parameter; that is,

$$u(x,y) = \begin{cases} b(x,y) & \text{if } (x,y) \in \Theta_1 \\ \sigma_4 b(x,y) & \text{if } (x,y) \in \Theta_2 \end{cases} \qquad \Theta_1 \cup \Theta_2 = \Theta$$

Here the following conditions are assumed to hold:

(a) The function b(x,y) is a quadratic polynomial in the local coordinate ζ on each 1-D domain in A, (see Section I.2 for the relevant mapping) .

(b) The intersection $\Theta_1 \cap \Theta_2$ may contain only 0-D domains with zero prescribed values, u(x,y) = 0

(c) If a 1-D domain is in some set Θ_i, (i=1,2), then the two adjacent 0-D domains also must be in the same Θ_i.

(d) If two or more 1-D domains with a common adjacent 0-D domain are in Θ_i, then the various b(x,y) functions defined on these 1-D domains must have an identical value at this 0-D domain.

The functions b(x,y), as quadratic polynomials in their local coordinates, are specified by their values at the 0-D domains, and at the mid-point of their 1-D domain (see Section II.2). The necessary quadratic interpolation for all other points of the 1-D domain is performed by the program.

With these specifications the finite element approximation of our given problem, on any admissible mesh, is a system of N nonlinear equations:

$$F_i(u) = \int_\Omega \left[\frac{\partial \Phi}{\partial U} v_i + \frac{\partial \Phi}{\partial U_x} \frac{\partial v_i}{\partial x} + \frac{\partial \Phi}{\partial U_y} \frac{\partial v_i}{\partial y} \right] dxdy$$

$$- \int_\Omega G_2 v_i \, dxdy - \int_\Gamma G_1 w_i \, d\gamma = 0 \quad , \quad i=1,2,...,N$$

where

$u = u(x,y)$ is the desired finite element approximation that satisfied the boundary conditions;

v_i is the global shape function corresponding to the i-th node of the (curvilinear) mesh on Ω;

Γ is a given subset of the union of 0-D domains and 1-D domains on which the Neumann boundary conditions specified by the third integral are defined;

w_i is, for any node on Γ, the induced one-dimensional quadratic shape function;

$\Phi = \Phi(\sigma_1,x,y,U,U_x,U_y)$, and $G_i = G_i(\sigma_i,x,y)$, i=1,2

are the given functions of the problem as discussed in Section I.1;

σ_i , i=1,2,3 are the two-dimensional parameter vectors and σ_4 the scalar parameter discussed in Section I.1. Recall from Section I.1, that for the computations the parameters σ_1, σ_2, σ_3 are specified as linear functions of two effective parameters λ_1, λ_2.

The Jacobian of the non-linear mapping $F = (F_1, F_2, ..., F_N)$ is easily obtained by direct differentiation. For ease of notation, let

$$u^0 = u \ , \ u^1 = \frac{\partial u}{\partial x} \ , \ u^2 = \frac{\partial u}{\partial y} \ , \ v_i^0 = v_i \ , \ v_i^1 = \frac{\partial v_i}{\partial x} \ , \ v_i^2 = \frac{\partial v_i}{\partial y} \ , \ i=1,2,...,N$$

Then the components of the Jacobian of F have the form

$$\frac{\partial F_i}{\partial u_j} = \int_\Omega \sum_{k,l=0}^{2} \frac{\partial^2 \Phi}{\partial u^k \partial u^l} \ v_i^k \ v_j^l \ dx \ dy \qquad , i,j=1,2,...,N$$

and, with the notation of Section I.1, the derivatives with respect to the effective parameters are

$$\frac{\partial F_i}{\partial \lambda_j} = \int_\Omega \sum_{k=0}^{2} \left[\beta_1^j \frac{\partial^2 \Phi}{\partial u^k \partial \sigma_1^j} + \beta_4^j \frac{\partial^2 \Phi}{\partial u^k \partial \sigma_4^j} \right] v_i^k \ dx \ dy$$

$$- \beta_2^j \int_\Omega \frac{\partial G_2}{\partial \sigma_2^j} v_i \ dx \ dy - \beta_3^j \int_\Gamma \frac{\partial G_1}{\partial \sigma_3^j} w_i \ d\gamma \qquad (i=1,2,...,N; \ j=1,2; \ \beta_4^1=\beta_4, \beta_4^2=0)$$

Here, it should be noted that, in the assembly of the function values and the Jacobian components, the fixed boundary nodes are not represented. Thus the derivatives with respect to σ_4 have to be transfered to the free nodes.

I.4. Mesh Representation and Densities

In NFEARS the admissible meshes on the basic square $S = \{(\xi,\eta),\ 0\leq\xi,\eta\leq 1\}$ are represented by a combination of tree structures. This representation is an extension of that given in [25] for linear elements. More specifically, the quadratic elements of the mesh in the 2-D domains are stored as extended quad-trees and their one-dimensional parts on the free 1-D domains as binary trees. This tree structure facilitates the required mesh-refinements and de-refinements. We refer to the cited article for a detailed description of the data structure and summarize here only some of the changes that were needed to accomodate quadratic elements.

As discussed in Section I.3 above, all meshes on Ω are first defined on the basic square S. Moreover, by definition of the admissible meshes, the initial mesh on each 2-D domain corresponds to a basic subdivision of S into four congruent squares, the superelements. Figure I.4.1 shows this initial subdivision with the four parts numbered 2 to 5. The mid-point carries the number 1 and the mid-points of the sides are assigned the indices 6 to 9. Further side nodes are required on the boundary of S but they are not numbered in the Figure. The superelements may be further refined but can never be de-refined; that is, each 2-D domain contains at least nine internal nodes.

The initial sub-division is represented by a labeled tree, where the root corresponds to the mid-point of the unit-square and the 8 descendant nodes to the four superelements and the four side-nodes, respectively. Labels l_x and l_y with values -1, 0 and +1 are assigned to the arcs of the tree to characterize the geometrical location of the corresponding node in relation to the mid-point. Note that the terminal nodes 2 to 5 represent mid-points of open squares, nodes 6 to 9 are mid-points of open intervals, and the root corresponds to a single point.

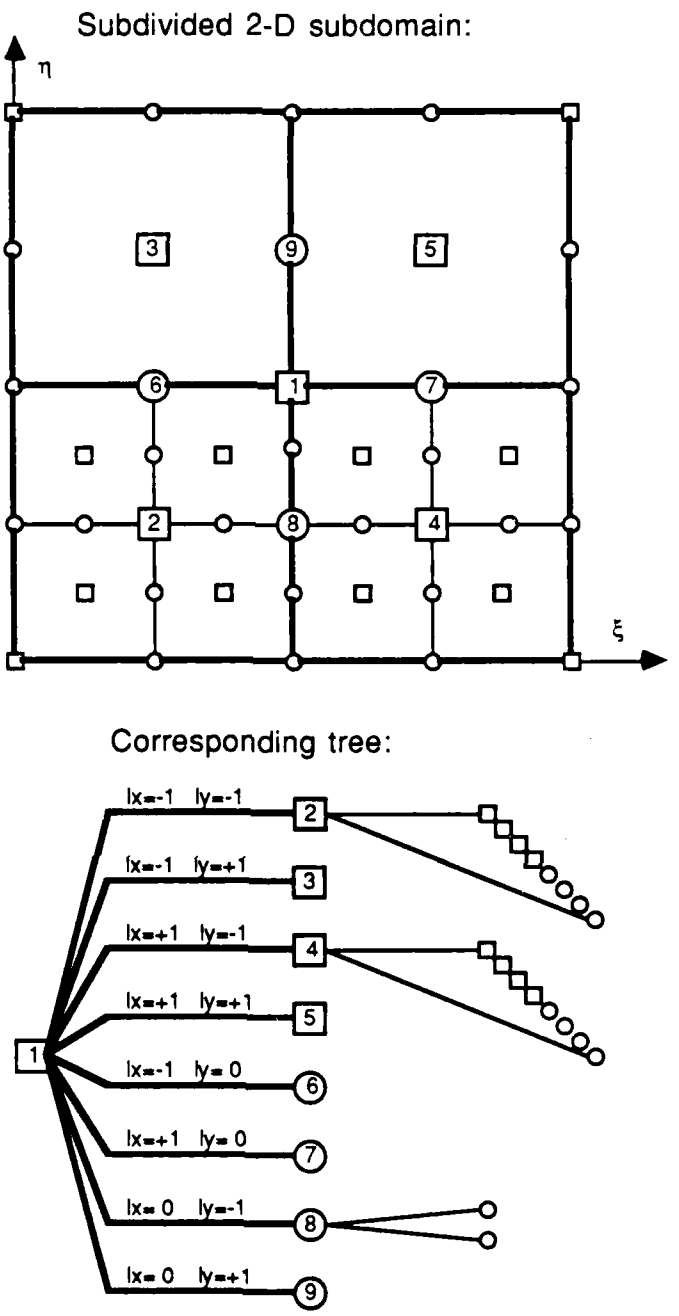


Figure I.4.1

The thin lines in Figure I.4.1 indicate the effect of a refinement of elements 2 and 4. Eight descendant nodes are appended to the particular tree nodes in correspondance with the four new elements and four new side-nodes. As in the case of the arcs from node 1, the eight arcs to the new nodes carry labels l_x, l_y

with values -1,0,+1. Two further side-nodes are introduced on both sides of node 8. These are implemented by adding two son-nodes to node 8 and labeling the arcs $l_x=0, l_y=-1$ (below 8) and $l_x=0, l_y=+1$ (above 8), respectively. In addition, so-called irregular nodes appear on both sides of nodes 6 and 7. Generally, a node of any mesh is irregular if it lies on the side of some element but is not the mid-point of that side. These irregular nodes are not represented in the tree. Solution values at the irregular points are linearly dependent on the solution values at the corner-points and at the mid-point of the side. Side points are also introduced on the bordering 1-D domains. Whether these are regular nodes or irregular nodes will depend on the mesh on the neighbouring 2-D domain, if there is one. If there is none; that is, if the 1-D domain is an outside boundary, then those nodes are defined to be regular nodes.

The resulting labeled tree has two important properties:

1. All non-terminal nodes of the tree represent single points while the terminal nodes correspond to the mid-points of open squares or open horizontal or vertical intervals in the local coordinate system. Open squares are mapped onto the interior of elements of Ω and the l_x, l_y labels on their incoming arcs carry non-zero values. Open intervals are those which have a label $l_x=0$ (vertical sides) or $l_y=0$ (horizontal sides).

2. There exists a unique path from the root to any given node on the tree which consists of arcs with labels l_{x_i}, l_{y_i} , $i=1,2,\ldots,m$, that define the local coordinates of the corresponding point as

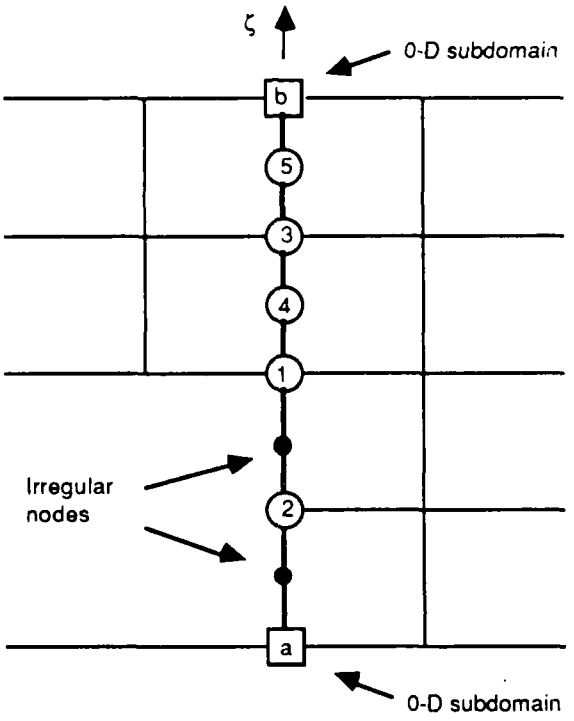
$$\xi = \frac{1}{2}\left(1 + \sum_{i=1}^{m} l_{x_i}\, 2^{-i}\right) \quad \text{and} \quad \eta = \frac{1}{2}\left(1 + \sum_{i=1}^{m} l_{y_j}\, 2^{-i}\right)$$

For terminal nodes the length m of the path specifies the length h of the side of the corresponding open square or interval as $h = 2^{-m}$.

The 1-D domains are mapped onto open unit intervals (see Section I.3). Each free 1-D domain is represented by a labeled binary tree, while fixed 1-D domains; that is, those that carry Dirichlet conditions, are only documented in summary records. Initially, a binary tree has one root-node and two son-nodes with arcs carrying the labels -1 and +1, respectively. The root node corresponds

to the mid-point of the 1-D domain and is also a cornerpoint of a superelement in the adjacent 2-D domains; the son-nodes are then representing the sides of this superelement. This is shown in Figure I.4.2.

1-D subdomain separating two 2-D subdomains:



Corresponding binary tree:

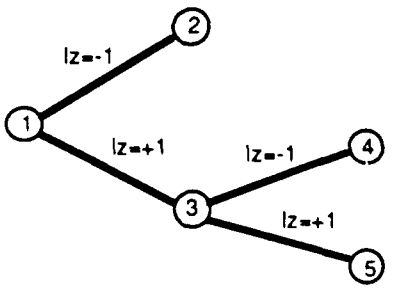


Figure I.4.2

For the control of the mesh modification process, as well as for the initial definition of the meshes on Ω, we follow [3], and introduce a density function and mesh intensity.

The 2-D domain Ω_k^2 is mapped onto a unit square which contains four superelements S_j^k, j=1,2,3,4. As in Section I.3 let $v_i(\xi,\eta)$, i=1,...,9, be the nine global shape functions of the superelement S_j^k. For every superelement exactly one corner point is the image of a 0-D domain. Suppose that this point corresponds to the corner with index I of the master element. Moreover, let $r_0(\xi,\eta)$ denote the distance between it and any point $(\xi,\eta) \in S_j^k$. Then the un-normalized density function d_j^k on S_j^k is defined by

$$\log(d_j^k(\xi,\eta)) = \sum_{i=1}^{9} p_i\, v_i(\xi,\eta) \;+\; 2\, p_0\, v_i(\xi,\eta)\, \log(r_0(\xi,\eta)) \,, \qquad \xi,\eta \in S_j^k$$

where $p_0,\ldots,p_9$ are given constants. This definition pre-supposes that the solution can be split into a smooth part and the effect of a singularity near the 0-D domain associated with S_j^k . More specifically, the sum over the shape-functions reflects mesh for the smooth part and the logarithmic term mesh for the singular part. Line singularities are not represented separately; that is, in our setting, they must be incorporated into the smooth part.

When a node belongs to two different superelements, then the coefficients p_i corresponding to it in these elements are assumed to be identical. With this a continuous, positive density function $d_k(\xi,\eta)$ is defined on the entire unit square. Moreover, $d_k(\xi,\eta)$ is uniquely characterized by 29 constants, namely, 25 coefficients $p_1,\ldots,p_9$ (nine on the open 2-D domain, three each on its four open 1-D domains, and one each at its four 0-D domains), and 4 singularity exponents p_0 (one each at the four 0-D domains).

The density functions d_k can be extended to a continuous, positive density function d on the entire domain Ω by requiring that the coefficients at any nodes in the intersection of two closed 2-D domains are identical. This function d is

completely characterized by nine coefficients for each open 2-D domain, three for each open 1-D domain, and two for each 0-D domain. Then

$$D_\Omega(\xi,\eta) = \frac{d(\xi,\eta)}{\sum_{k=1}^{N_2} \int_{S^k} d_k(\xi,\eta)\, d\xi\, d\eta}$$

shall be the normalized density function on the domain Ω.

Let Δ_j, $j = 1,2,...$, be a sequence of meshes with E_j elements such that $E_j \to \infty$ as $j \to \infty$. Moreover, assume that for any open subset Ω' of Ω which contains $E_{j\Omega'}$ (open) elements, we have the inequality

$$\alpha_1(j,\Omega) \int_\Omega D_\Omega(\xi,\eta)\, d\xi\, d\eta \le \frac{E_{j\Omega'}}{E_j} \le \alpha_2(j,\Omega') \int_{\Omega'} D_\Omega(\xi,\eta)\, d\xi\, d\eta$$

where

$$\lim_{j\to\infty} \sup\, \alpha_2(j,\Omega) \qquad \text{and} \qquad \lim_{j\to\infty} \inf\, \alpha_1(j,\Omega)$$

are finite and independent of Ω'. Then, evidently, for large j the quantity

$$\Im(\Omega') = \frac{1}{E_{j\Omega'}} \int_{\Omega'} D_\Omega(\xi,\eta)\, d\xi\, d\eta$$

is nearly independent of Ω'. This suggests that we may characterize a mesh by its normalized density function and a number $\Im$, called its intensity, such that

$$E_{\Omega'} = \frac{1}{\Im} \int_{\Omega'} D_\Omega(\xi,\eta)\, d\xi\, d\eta$$

is the number of elements in any open subset Ω' of Ω. This relation allows us to construct an admissible mesh Δ on Ω once a normalized density function $D_\Omega(\xi,\eta)$ and an intensity $\Im$ are known. Clearly, it suffices to give this algorithm

only for some closed 2-D domain Ω_k^2 of Ω . As before, let Ω_k^2 be mapped onto the unit square S^k and let Δ_k be the mesh consisting of the four superelements S^k.

For all elements ω of Δ_k do
    Compute $E_{\Omega'}$ for the set $\Omega' = \{\omega\}$;
    If $E_{\Omega'} > 1$ then
        construct a new mesh Δ_k by subdividing ω into four new elements,
    else
        terminate the process with the current mesh Δ_k.

The process ends with a mesh Δ_k for which $E_{\Omega'} \leq 1$ for all singleton sets $\Omega' = \{\omega\}$ of Δ_k. Moreover, we also know that $E_{\Omega'} > 1$ for the father ω' of ω; that is, for the element of an earlier mesh from which ω was constructed by the quartering rule.

Some examples of this process are given in Figure II.3.4 . It may be noted that there the intensity was set equal to 1/20, and hence we should expect a mesh with 20 elements. But the actual number of elements is closer to 50 due to our use of the quartering rule; that is, the requirement that only admissible meshes are allowed.

I.5. The Solution Manifold

In Sections I.3 we specified our problem in the form of a system of nonlinear equations

$$F(u, \lambda_1, \lambda_2) = 0 \quad , \quad F: R^N \times R^2 \rightarrow R^N,$$

involving the vector u of the unknown values at the N free nodes of the finite element mesh on Ω and the two effective parameters λ_1, λ_2. This system approximates the nonlinear operator equation introduced in Section I.1, and, as indicated there, the set of all solutions $(u,\lambda_1,\lambda_2) \in R^N \times R^2$ forms a differentiable manifold in $R^N \times R^2$. We summarize here briefly some of the necessary differential geometric properties of this solution manifold and refer for further details, for instance, to [19] and the literature cited there.

For ease of notation, we shall use the abbreviation $\underline{x} = (u, \lambda_1, \lambda_2) \in R^{N+2}$ and write $DF(\underline{x})$ for the N x (N+2) Jacobian of F. A point $\underline{x}$ is called regular if the Jacobian has full rank N. Then the set of all regular solutions

$$M = \{ \underline{x} \in R^{N+2};\ F(\underline{x}) = 0,\ \text{rge}DF(\underline{x}) = N \}$$

is a two-dimensional differentiable manifold in R^{N+2} of the same differentiability class as F. By restricting ourselves to this regular solution manifold we asume that a suitable unfolding of the original problem has been chosen. In practical applications, this is, in general, a natural assumption.

As indicated already in Section I.1, we wish to analyse the shape and characteristic features of this solution manifold. However, for this we have to recall that the mapping F represents only a discrete approximation of the original (infinite) dimensional problem of Section I.1. As noted there, this original problem -- after restriction to the two effective parameters λ_1, λ_2 -- may also be expected to possess a two-dimensional solution manifold in the product of the trial space X and the parameter space R^2. It is actually this original manifold which we wish to analyse; but, of course, that manifold is not directly accessible for a computational study. This raises the question of the influence of the finite element discretization upon the shape and features of the original

manifold and of estimates of the size of the discretization error. This will be discussed in Section I.8 below. Here we concentrate on the above manifold M defined by the specific finite element discretization F.

At any point $\underline{x} \in M$ the tangent space $T_{\underline{x}}M$ of M may be identified with the null-space of the Jacobian; that is,

$$T_{\underline{x}}M = \ker DF(\underline{x}) = \{ w \in R^{N+2},\ DF(\underline{x})w = 0 \}.$$

Accordingly, the orthogonal complement

$$N_{\underline{x}}M = (T_{\underline{x}}M)^{\perp} = \mathrm{rge}\ DF(\underline{x})^T$$

is the normal space at that point.

For the analysis of any differentiable manifold we require suitable local coordinate systems on M. For the computations we use local coordinate systems induced by given 2-dimensional subspaces T of R^{N+2}. More specifically, any such space T induces a local coordinate system of M at $x \in M$ provided that

$$T \cap T_{\underline{x}}M = \{0\}.$$

If this condition holds then there exist open neighborhoods V_1 and V_2 of the origins of T and R^{N+2}, respectively, as well as a unique differentiable function w: $V_1 \rightarrow T^{\perp}$, $w(0) = 0$, such that

$$M \cap V_2 = \{ \underline{y} \in R^{N+2};\ \underline{y} = \underline{x} + t + w(t),\ t \in V_1 \}.$$

A point $\underline{x}$ is a non-singular point with respect to the given coordinate space T if T induces a local coordinate system of M at $\underline{x}$, otherwise it is called a singular point or foldpoint.

Let Λ be the two-dimensional subspace of R^{N+2} spanned by the effective parameter directions. Then interest centers especially on determining the foldpoints with respect to the coordinate space $T = \Lambda$. If our original problem

concerns a study of equilibrium configurations of a mechanical structure then these foldpoints with respect to Λ represent approximations to the eqilibria where a change of the stability behavior of the structure may be expected occur.

The basic procedures for the computational analysis of our solution manifold M are the continuation methods. These methods require a restriction to some path on M and then produce a sequence of points along that path. One such continuation procedure is incorporated in NFEARS. The allowable paths for this procedure are defined by specifying the effective parameters λ_1, λ_2 as linear functions of a scalar parameter λ:

$$\lambda_i = \delta_i \lambda \ , i=1,2 , \quad \delta_1 + \delta_2 = 1.$$

Some of the details of the continuation process and of its capabilities are discussed in Section I.6.

Obviously, it is not easy to develop a good picture of a two-dimensional manifold solely from information along such paths. This led recently to the development of methods for the computation of simplicial approximations of two-dimensional open subsets of M. NFEARS incorporates a form of the method introduced in [21],[22] which will be discussed in Section I.7 below.

I.6. The Continuation Process

The continuation method incorporated into NFEARS follows the design of the PITCON system (see [23],[24]) . We summarize here only the principal aspects of that method and refer to the original articles for further detail.

As discussed in Section I.3, our finite-element approximation has the form

$$F_i(u, \lambda_1, \lambda_2) = 0 , \quad i=1,2,...N, \quad u \in R^N.$$

Let M again be the corresponding regular solution manifold and, for ease of notation, write $\underline{x} = (u, \lambda_1, \lambda_2)$. For the continuation process we have to restrict ourselves to a path on M through a specified point $\underline{x}^0 = (u^0, \lambda_1^0, \lambda_2^0) \in M$. This is

equivalent with an augmentation of the system by some scalar equation. As mentioned in Section I.5, NFEARS uses only paths on M which are specified by a linear combination of the effective parameters λ_1, λ_2. In other words, we consider augmented systems of the form

$$F^*(u,\lambda_1,\lambda_2) = \begin{bmatrix} F_1(u,\lambda_1,\lambda_2) \\ \cdots\cdots\cdots\cdots \\ F_N(u,\lambda_1,\lambda_2) \\ \delta_1(\lambda_1-\lambda_1^0) + \delta_2(\lambda_2-\lambda_2^0) \end{bmatrix} = 0$$

where δ_1, δ_2 are the given constants introduced at the end of Section I.5. Let

$$\Pi = \{ \underline{x} \in R^{N+2}, F^*(\underline{x}) = 0, \text{ rge } DF^*(\underline{x}) = R^{N+1} \}$$

be the regular solution manifold of the augmented equations. The connected component Π_0 of Π containing the given initial point $\underline{x}^0$ is the path on M which is to be computed.

The continuation process begins from the given starting point $\underline{x}^0$ and produces a sequence of points $\underline{x}^k$, k=0,1,... , which approximate points of M. For any $k \geq 0$

the step from $\underline{x}^k$ to $\underline{x}^{k+1}$ corresponds to an implementation of the local coordinate representation discussed in Section I.4. More specifically, if T=span{t}, with $t \in R^{N+2}$, $t \neq 0$, is a local coordinate space of M at $\underline{x}^k$, then, for any fixed $\underline{y} \in R^{N+2}$, the Jacobian of the augmented equations

$$\begin{bmatrix} F^*(\underline{x}) \\ t^T(\underline{x} - \underline{y}) \end{bmatrix} = 0$$

is non-singular for all $\underline{x}$ in some neighborhood of $\underline{x}^k$. Thus if $\underline{y}$ approximates a point of Π in that neighborhood, then it follows readily that the above system has a unique solution $\underline{x}^{k+1} \in M$ which can be computed by means of a locally convergent iterative process, started, say, at $\underline{y}$.

This gives an outline of the process and it remains only to specify the particular choices used in PITCON and NFEARS. At $\underline{x}^k$ the normalized tangent vector $\underline{w}^k$ of Π is computed; that is,

$$DF^*(\underline{x}^k)\underline{w}^k = 0 \ , \quad ||\underline{w}^k|| = 1,$$

where for k=0 the direction of $\underline{w}^k$ is user-given and for k>0 it is defined by

$$(\underline{w}^k)^T (\underline{x}^k - \underline{x}^{k-1}) > 0.$$

As a check against leaving the connected component Π_0 the condition

$$\det(DF^*(\underline{x}^k)^T, \underline{w}^k) > 0$$

is also monitored. The predicted point $\underline{y}$ is now computed by linear extrapolation along the tangent direction; that is,

$$\underline{y} = \underline{x}^k + \tau \underline{w}^k.$$

The choice of the step length τ is the same as in PITCON and we refer to [23] for the detailed discussion.

For the definition of the local coordinate system $T = \text{span}\{t\}$ at $\underline{x}^k$ one of the natural basis vectors of R^{N+2} is used; that is we set $t = \underline{e}^i$ where the index i is chosen such that $|(\underline{w}^{k)T}\underline{e}^i|$ is maximal. This defines the above given augmented system which is then solved by means of a chord Newton process started from the predicted point $\underline{y}$. For details of the acceptance and rejection tests of this iterative process and for the recovery procedures after failure we again refer to the cited article [23].

The process allows for the computation of target points; that is of points $\underline{x} \in M$ where the component $\underline{x}^T e^j$ with a prescribed index j has a specified value ξ^*. In order to detect such a point the condition

$$\text{sign}((\underline{x}^{k-1})^T e^j - \xi^*) \neq \text{sign}((\underline{x}^k)^T e^j - \xi^*)$$

is monitored. If it holds, the interpolated point

$$\underline{y}^* = (1-\upsilon)\underline{x}^{k-1} + \upsilon\, \underline{x}^k \ , \quad (e^j)^T \underline{y}^* = \xi^*$$

is evaluated and then the corrector process is applied with the augmenting equation $(e^j)^T\underline{x} - \xi^* = 0$ and with $\underline{y}^*$ as starting point. Once again we refer to [23] for further details.

Finally, the continuation process incorporates a procedure for computing limit points of Π with respect to a specified natural basis direction e^j; that is of points $\underline{x}$ on Π where the j-th component of the tangent vector is zero. The computation of these points is similar to that of the target points. For their detection the condition

$$\text{sign}((\underline{w}^{k-1})^T e^j) \neq \text{sign}((\underline{w}^k)^T e^j)$$

is monitored. If it holds, then the limit point process of [23] in the form discussed in [15] is applied. We refer to these papers for the details.

I.7. Simplicial Approximations of the Solution Manifold

As observed already earlier, it is difficult to gain a picture of the two-dimensional solution manifold M of our nonlinear equations

$$F(\underline{x}) = 0 \quad , \quad \underline{x} \in R^{N+2}$$

solely from the information along the paths used by the continuation process. For this purpose a form of the method described in [21], [22] for computing simplicial approximations of open regions of M was incorporated in NFEARS. Once again we refer to the original articles for details and concentrate mainly on the features that are different.

The approximation process begins with the choice of a reference mesh Δ of R^2. In the original method a equilateral triangular mesh was used while NFEARS employs a planar Kuhn-triangulation Δ as shown in Figure I.7.1.

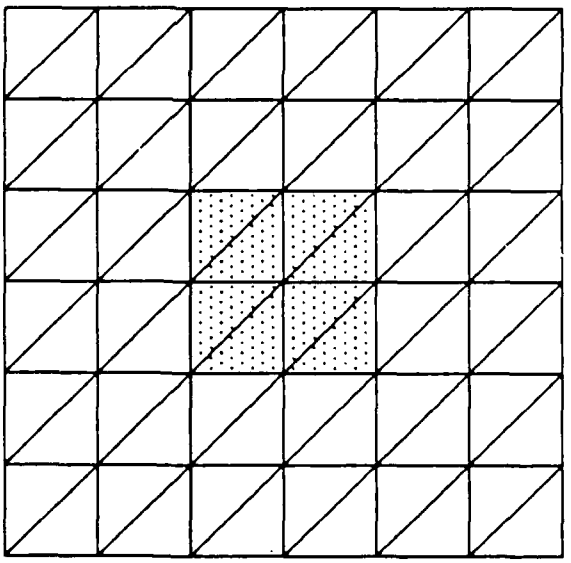


Figure I.7.1

The aim of the triangulation process is to transfer the nodes of a part of Δ, together with their connnectivity information, from R^2 onto M. This transfer process works with groups of nodes. More specifically, in NFEARS, we use "patches" of eight triangles which form rectangles of Δ and contain nine nodes,

namely, one center node and eight boundary nodes. Figure II.6.1 shows a portion of such a triangulation with one patch of triangles indicated as a hatched rectangle.

The continuation process is used to generate a first "center" point $\underline{x} \in M$ for the "region" on M that is to be triangulated. Now a patch of the reference mesh Δ is mapped linearly onto the affine tangent space $\underline{x}+T_{\underline{x}}M$ in such a way that the center point of the patch corresponds to x. For this mapping we use an appropriate basis w_1, w_2 of $T_{\underline{x}}M$ together with two user-defined scale factors τ_1, τ_2 for each one of these directions. In other words, the images of the nine nodes of the patch on $\underline{x}+T_xM$ are

$$x + k_1 \tau_1 w_1 + k_2 \tau_2 w_2 , \quad k_1 = -1,0,1 , \quad k_2 = -1,0,1$$

These points are now projected onto M. In the original method in [21] , [22] a chord Gauss-Newton method is used to produce a projection from $\underline{x}+T_{\underline{x}}M$ onto M which is orthogonal to the tangent space. In order to retain the same data structure as in the continuation process, we use in NFEARS the chord-Newton process applied to the augmented equations

$$\begin{bmatrix} F(\underline{z}) \\ (e^{i})^{T}(\underline{z} - \underline{y}) \\ (e^{j})^{T}(\underline{z} - \underline{y}) \end{bmatrix} = 0$$

Here i and j are the indices of the largest components in modulus of w_1 and w_2, respectively, and $\underline{y} \in \underline{x}+T_{\underline{x}}M$ is the point which is to be "projected" onto M.

In order to repeat these steps with a neighboring patch, a new center point on M is constructed by projecting an appropriate point along one of the basis lines of $\underline{x}+T_{\underline{x}}M$ onto M. More specifically, if, in Δ, the new patch is located in the "upward", "left", "right",or "downward" direction from the original one, then the new center point is the projection onto M of the points $\underline{x}+2\tau_2 w_2$, $\underline{x}-2\tau_1 w_1$, $\underline{x}+2\tau_1 w_1$, $\underline{x}-2\tau_2 w_2$, respectively. Now a suitable basis of the tangent space of M at the new center point has to be constructed which then allows us to map the new patch onto M (using the same scale factors τ_1, τ_2 as before). But, of course, nodes of Δ that have already been mapped onto M will not be used again.

Clearly, in order to ensure that the images of the rectangles of Δ on M form a curvi-linear mesh on the manifold, we need to use bases on the tangent spaces $T_x M$ which change smoothly from point to point. In the terminology of differential geometry this means that we require an algorithm which generates a moving frame on the portion of M under consideration. Standard methods for computing tangent bases do not produce continuously varying tangent bases (see e.g. [11]). A slightly modified form of the moving frame algorithm introduced in [21] is used in NFEARS to construct the basis vectors of the tangent spaces at the center points of the patches.

Suppose that $\underline{x} \in M$ is a point where a basis of the tangent space $T_{\underline{x}} M$ is to be computed. If we choose two distinct natural basis vector e^i and e^j of R^{N+2}, neither of which is orthogonal to $T_{\underline{x}} M$, then the solutions q_1, q_2 of the augmented system

$$\begin{bmatrix} DF(\underline{x}) \\ (e^i)^T \\ (e^j)^T \end{bmatrix} q_k = e^{N+k}, \quad k=1,2$$

certainly form a basis of $T_{\underline{x}} M$. Accordingly, the vectors t_1, t_2

$$t_1 = \frac{q_1}{\| q_1 \|_2}, \quad t_2 = \frac{1}{\chi} \left[q_2 - \left((q_2)^T q_1 \right) q_1 \right], \quad \chi = \| q_2 - \left((q_2)^T q_1 \right) q_1 \|_2$$

represent an orthogonal basis of $T_{\underline{x}} M$. As in the continuation process, the indices of the largest components in modulus of t_1 and t_2 can be used to determine the indices i and j of the basis vectors in the augmented system.

Once an orthogonal basis t_1, t_2 of $T_{\underline{x}} M$ has been obtained the quantities

$$\theta_k = \cos(\vartheta_k) = \sqrt{((e^k)^T t_1)^2 + ((e^k)^T t_2)^2}, \quad k = 1,2,\ldots,N+2$$

are the direction cosines of the principal angles between $T_{\underline{x}}M$ and the natural basis vector of R^{N+2}. Let i_1, i_2 be the indices corresponding to the largest of these θ_k, with ties broken by lexicographic ordering. Then we form the matrix

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} , \quad a_{jk} = (e^{i_j})^T t_k , \quad j,k = 1,2$$

and with it the quantities

$$\chi^1 = \frac{1}{\chi^d}(a_{11} + a_{22}) , \quad \chi^2 = \frac{1}{\chi^d}(a_{12} - a_{21}), \quad \chi^d = \sqrt{(a_{11} + a_{22})^2 + (a_{12} - a_{21})^2}$$

It was shown in [22], that the vectors w_1, w_2

$$w_1 = \pm(\chi^1 t_1 - \chi^2 t_2) , \quad w_2 = \pm(\chi^2 t_1 + \chi^1 t_2)$$

with the signs chosen such that

$$(e^{i_k})^T t_k > 0 , \quad k=1,2$$

provide the desired "moving frame" basis of T_xM. Evidently, only the original basis vectors q_1 , q_2 together with the various transformation coefficients have to be retained and the two sets of w-vectors need never be computed explicitly.

The sequencing of the "transfer" of the patches of Δ onto M is controlled by the user and will be described in Section II.6.

I.8. Error Estimation and Mesh Adaptation

Let Δ be a given mesh on Ω and u the computed finite element solution on Δ. The calculation of the a posteriori error estimate $\varepsilon(\Delta)$ of u follows the approach for linear problems discussed, for instance, in the surveys [2], [7], [8], [10] and in [3], [9]. In particular, $\varepsilon(\Delta)$ is obtained as a sum of squares of error indicators $\rho(\omega)$ of the elements ω of Δ; that is,

$$\varepsilon(\Delta)^2 = \sum_{\omega \in \Delta} \rho(\omega)^2$$

These error indicators are calculated in a two-step process. First, it is assumed that the solution is sufficiently smooth and hence that the indicators of neighboring elements do not differ too much. Then this assumption is checked and, if the indicators of neighboring elements (of a common father node in the quad-tree of Section I.4) are different, then their computed indicators are adjusted accordingly. This adjustment turns out to be desirable since the unmodified error indicators under-estimate the error near singularities.

In addition to the quadratic shape functions of Section I.3 we introduce on the master element of Figure I.3.1 the error shape functions

$$\psi_1(\mu,\nu) = \nu\,(\nu^2 - 1)\,(\mu^2 - 1)\,, \quad \psi_2(\mu,\nu) = \mu\,(\nu^2 - 1)\,(\mu^2 - 1)$$

On any element ω, this induces the global error shape functions, $w_i(x,y) = \psi_i(\mu(x,y),\nu(x,y))$, i=1,2, and with

$$R_k = \frac{\int_\omega \frac{\partial^2 \Phi}{\partial u^{k\ 2}}\, dx\, dy}{\int_\omega dx\, dy}\,, \qquad k=1,2$$

$$Q_k = \int_\omega \Big(\sum_{i,j=0}^{2} \frac{\partial^2 \Phi}{\partial u^i\, \partial u^j}\, u^i\, w_k^j - G_2(\sigma_2,x,y)\, w_k \Big)\, dx\, dy\,, \quad k=1,2$$

the first approximation of the error indicator on ω is defined by

$$\rho'(\omega) = 720\left[\frac{Q_1^2}{R_1^2} + \frac{Q_2^2}{R_2^2}\right]$$

In addition, NFEARS calculates a linear energy term for ω in the form

$$En\ (\omega) = \int_{\omega} \sum_{i,j=0}^{2} \frac{\partial^2 \Phi}{\partial u^i\, \partial u^j}\, u^i\, u^j\, dx\, dy$$

Now, an adjusted error indicator $\rho = \rho(\omega)$ is calculated from ρ' for those elements ω whose "brothers" are also elements of the current mesh Δ. In other words, if ω' is the father of ω then all four "sons" of ω' must also be elements of Δ. For instance, in Figure I.8.1, only the elements ω_i, i=1,2,3 and 4, are candidates for adjustment, while the elements marked by ω_0 retain their already calculated error indicators ρ'; that is, we set $\rho(\omega_0) = \rho'(\omega_0)$.

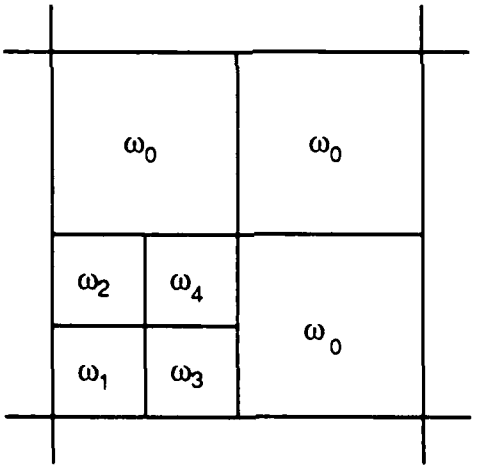


Figure I.8.1

Any such set of four elements $\omega_1,\ldots,\omega_4$ of Δ will be called an adjustable cluster. The adjustment of the error indicators of such a cluster proceeds as follows: We order the four elements in decreasing order of their calculated indicators $\rho_i' = \rho'(\omega_i)$, say,

$$\rho_1' \geq \rho_2' \geq \rho_3' \geq \rho_4'$$

and then define the ratio between the two largest error indicators and those of the corresponding, diagonally opposite elements ω_1^d and ω_2^d:

$$r_1 = \frac{\rho_1'}{\rho'(\omega_1^d)} \qquad \text{and} \qquad r_2 = \frac{\rho_2'}{\rho'(\omega_2^d)}$$

Now the adjusted error indicators ρ are calculated by the algorithm:

If $\rho_1' > 8\,\rho_2'$
  then $\rho_1 = g_1(r1)$, $\rho_k = \rho_k'$,k=2,3,4
else if $\rho_2' > 100\,\rho_3'$
  then $\rho_1 = g_2(r_1)$, $\rho_2 = g_2(r_2)$, $\rho_3 = \rho_3'$, $\rho_4 = \rho_4'$
  else $\rho_k = \rho_k'$,k=1,2,3,4

Here the two function $g_1(r)$ and $g_2(r)$ are defined by

$$g_1(r) = \begin{cases} 3.4813\ e^{0.003762r} & \text{if } r > 234 \\ 0.031605\,r + 1 & \text{if } r \leq 234 \end{cases}$$

$$g_2(r) = \begin{cases} 1.7132\ e^{0.001\,r} & \text{if } r > 700 \\ 0.0034999\,r + 1 & \text{if } r \leq 700 \end{cases}$$

Let $h=h(\omega)$ be the side-length of the image of the element ω in the unit-square S. Then the computed error indicator $\rho(\omega)$ of ω can be shown to have the following relation to the $H^3(\omega)$-norm of the exact solution:

$$\rho^2(\omega) = C \,\| u \|^2_{H^3(\omega)} h^4(\omega)$$

This suggests that we introduce the quantity

$$\tau(\omega) = \frac{\rho(\omega)}{h(\omega)^3}$$

for which

$$\rho(\omega)^2 = h(\omega)^4 \int_{\omega} \tau(\omega)^2 \, d\xi \, d\eta$$

in the local coordinates (ξ,η) on the unit square S.

Let S^k be the unit square containing the element ω and suppose that a normalized density function $D_\Omega(\xi,\eta)$ and intensity $\mathfrak{I}$ are given. Then the size h of an element ω containing the point ξ,η is asymptotically equal to

$$h(\omega) = \frac{\mathfrak{I}^{1/2}}{D_\Omega^{1/2}(\xi,\eta)}$$

which leads to the asymptotic error-indicator formula

$$\rho''(\omega)^2 \approx \mathfrak{I}^2 \int_{\omega} \frac{\tau(\omega)^2}{D_\Omega^2} \, d\xi \, d\eta \ , \qquad \varepsilon(\Delta)^2 = \sum_{\omega \in \Delta} \rho''(\tau)^2$$

For a minimal error, this formula suggests that we should choose the density function $D_\Omega(\xi,\eta)$ so as to minimize the expression

$$\sum_{k=1}^{N} \int_{S^k} \frac{\tau(\omega)^2}{D_\Omega^2} \, d\xi \, d\eta$$

This minimization is performed on the superelement S^k to which ω belongs. The definition of the unnormalized density function d_j^k on a superelement S_j^k is given in Section I.4. The desired coefficients p_0 and p_i , i=1,...,9 are calculated by a least square fit. More specifically, for any element ω in S_j^k with side-length $h(\omega)$ we introduce the least squares functional

$$\sum_{\omega \in S_j^k} \int_{\omega} \left[\log(d_j^k(\xi,\eta)) - \log \frac{\rho^{2/3}(\omega)}{h^2(\omega)} \right] d\xi \, d\eta$$

which is minimized to calculate p_0 and $p_1,...,p_9$.

The resulting (raw) density functions d_j^k over the superlements of Ω are then adjusted to produce a continuous function over the full domain. If a node occurs as point i, $(1 \le i \le 9)$, in one superelement S^k and as point j, $(1 \le j \le 9)$, in another superelement $S^{k'}$, and if $p_i(\omega)$ and $p_j(\omega')$ are the corresponding coefficients of

the density in ω and ω', respectively, then the values of these two coefficients are replaced by their maximum; that is, we set

$$p_i(S_k) := p_j(S^{k'}) := \max \{ p_{min}, p_i(S^k), p_j(S^{k'}) \}$$

with a suitable minimal value p_{min} . Similarly we adjust the coefficients $p_0 = p_0(S^k)$, but now we use minima. In other words, if a particular 0-D domain is a corner point of the superelements S^k and $S^{k'}$ then we set

$$p_0(S^k) := \min \{ p_{max}, p_0(S^k), p_0(S^{k'}) \}$$

These adjusted p_0–coefficients are associated with the four 0-D domains of S and with this the calculation of the unnormalized density function $d_\Omega(\xi,\eta)$ is complete. The un-normalized density function is normalized again as described in Section I.4:

$$D_\Omega(\xi,\eta) = \frac{d_\Omega(\xi,\eta)}{\sum_{k=1}^{N_2} \int_{S^k} d_\Omega(\xi,\eta)\, d\xi\, d\eta}$$

With this adjustment, each closed 2-D domain has, as required, exactly 29 associated coefficients (see Section I.4). Accordingly, as discussed above, we can calculate on Ω the error indicator

$$\varepsilon_\Omega^2 = \Im \sum_{k=1}^{N_2} \sum_{\omega \in S_k} \int_{\omega} \frac{\rho(\omega)^2}{h(\omega)^6\, D_\Omega^2}\, d\xi\, d\eta$$

where the average intensity $\Im$ should be chosen such that $\varepsilon_\Omega \le \varepsilon_{max}$ with an a priori given tolerance ε_{max}. This defines the properties of the ideal mesh that should be used.

In practice we proceed analogously as in Section I.4. For each super-element S_j^k, there is a sub-tree in the quadtree representation of the current mesh Δ. Let ω' be a node of the sub-tree which again defines a sub-tree. Let now $A(\omega)$ be

the collection of all elements corresponding to the terminal nodes of this sub-tree. Then we can define the intensity

$$\Re(\omega') = \sum_{\omega \in A(\omega)} \int_{\omega} D_{\Omega}(\xi,\eta)\, d\xi\, d\eta$$

for ω'. With this $\Re(\omega')$ and the required intensity $\Im$, the modification of the current mesh is then based on the following two rules for each 2-D domain Ω_k^2 :

(a) If $\Re(\omega') > \Im$ and ω' is a father node of an element ω in the current mesh, then the sub-division of ω' is eliminated; that is, ω and its brothers are removed and ω' becomes an element in a de-refined mesh.

(b) If $\Re(\omega') < \Im$ and ω' is an element of the current mesh refined by subdividing ω' into four elements.

When a father node ω' is de-refined and, by rule (a), becomes an element, then rule (a) has to be applied again to its father node. Similarly, when, by rule (b), an element ω' is sub-divided then the new elements must also be checked by rule (b). NFEARS performs first the de-refinement with rule (a) until no further mesh modification occurs, then it proceeds with the refinement by rule (b), and the process ends when no more refinement is needed.

I.9 References

[1] I.Babuska, Feedback, Adaptivity, and A Posteriori Estimates in Finite Elements: Aims, Theory, and Experience; in [10], pp3-23

[2] I.Babuska,J.Chandra, J.E.Flaherty, editors, Adaptive Computational Methods for Partial Differential Equations, SIAM Publications, Philadelphia, PA 1983

[3] I.Babuska,W. Gui, Basic Principles of Feedback and Adaptive Approaches in the Finite Element Method, Comp. Meth. in Appl. Mech. and Eng. 55, 1986,27-42

[4] I.Babuska, A.Miller, M.Vogelius, Adaptive Methods and Error Estimation for Elliptic Problems of Structural Mechanics, in [2], pp57-73

[5] I.Babuska, W.C.Rheinboldt, Error Estimates for Adaptive Finite Element Computations; SIAM J. Num. Anal. 15, 1978, 736-754

[6] I.Babuska, W.C.Rheinboldt, Computational Error Estimates and Adaptive Processes for Some Nonlinear Structural Problems, Comp. Meth. in Appl. Mech. and Eng. 34, 1982, 895-937

[7] I.Babuska, W.C.Rheinboldt, A Survey of A Posteriori Error Estimators and Adaptive Approaches in the Finite Element Method; in "Proc. of the China-France Symp. on Finite Element Methods" ed. by Feng Kang and J.L.Lions, Gordon and Breach, Inc. New York, 1983 pp1-56

[8] I.Babuska, W.C.Rheinboldt, A Posteriori Error estimates and Adaptive Techniques for the Finite Element Method, in "Elliptix Problem Solvers" ed. by G.Birkhoff and A.Schoenstadt, Academic Press, New York 1984, pp345-378

[9] I.Babuska, D.Yu , Asymptotically Exact A Posteriori Error Estmator for Biquadratic Elements, Univ. of Maryland, Inst.of Phys. Science and Techn., Techn. Note BN-1050, June 1986

[10] I.Babuska, O.C.Zienkiewicz, J.Gago, E.R.de A.Oliveira, editors, Accuracy Estimates and Adaptive Refinements in Finite Element Computations, J.Wiley and Sons, New York 1986

[11] T.F.Coleman, D.C.Sorensen, A Note on the Computation of an Orthonormal Basis for the Null Space of a Matrix; Math. Prog. 29, 1984, 234-242

[12] J.P.Fink, W.C.Rheinboldt, On the Discretization Error of Parametrized Nonlinear Equations; SIAM J. Num. Anal. 20,1983,732-746

[13] J.P.Fink, W.C.Rheinboldt, Solution Manifolds of Parametrized Equations and their Disvretization Error; Num. Math. 45, 1984, 323-343

[14] J.P.Fink, W.C.Rheinboldt, Local Error Estimates for Parametrized Nonlinear Equations; SIAM J. Num. Anal. 22, 4, 1985

[15] R.G.Melhem, W.C.Rheinboldt, A Comparison of Methods for Determining Turning Points of Nonlinear Equations, Computing 29,1982,210-226

[16] C. Mesztenyi, A.Miller, W.Szymczak, FEARS: Details of Mathematical Formulation; Univ. of Maryland, Inst.of Phys. Science and Techn., Techn. Note BN-994, Dec. 1982

[17] C. Mesztenyi, W.Szymczak, FEARS User's Manual for UNIVAC 1100, Univ. of Maryland, Inst.of Phys. Science and Techn., Techn. Note BN-991, Oct. 1982

[18] W.C.Rheinboldt, Error Estimates for Nonlinear Finite Element Computations; Comp. and Struct. 20, 1985, 91-98

[19] W.C.Rheinboldt, Numerical Analysis of Parametrized Nonlinear Equations; J.Wiley and Sons,Inc., New York,NY 1986

[20] W.C.Rheinboldt, Error Estimations and Adaptive Techniques for Nonlinear Parametrized Equatuions; in [10], pp163-180

[21] W.C.Rheinboldt, On a Moving Frame Algorithm and the Triangulation of Equilibrium Manifolds, in "Bifurcations: Analysis, Algorithms, Applications" ed. by T.Kuepper, R.Seydel, H.Troger, ISNM Vol. 79, Birkhauser Verlag, Basel 1987, pp256-267

[22] W.C.Rheinboldt, On the Computation of Multi-Dimensional Solution Manifolds of Parametrized Equations, Univ. of Pittsburgh, Inst. f. Comp. Math. and Appl., Techn. rep. ICMA-86-102, Nov. 1986; Numer. Math., in press

[23] W.C.Rheinboldt, J.V.Burkhardt, A Locally Parametrized Continuation Process, ACM Trans. on Math. Software 9,1983,215-235

[24] W.C.Rheinboldt, J.V.Burkhardt, Algorithm 596: A Program for a Locally Parametrized Continuation Process, ACM Trans. on Math. Software 9,1983,236-246

[25] W.C.Rheinboldt, C.K.Mesztenyi, On a Data Structure for Adaptive Finite Element Mesh Refinements, ACM Trans. on Math. Software, 6, 1980, 166-187

[26] P.Zave, W.C.Rheinboldt, Design of an Adaptive, Parallel Finite-Element System;
ACM Transactions on Math. Software 5,1979,1-17

END

DATE

FILMED

5-88

DTIC